

DETAILED ACTION

Examiner's Amendment

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.
2. Authorization for this examiner's amendment was given in a telephone interview with Lawrence J. Merkel (Reg. No. 41,191) on January 15, 2010.
3. The application has been amended as follows:
 1. (Cancelled)
 2. (Currently Amended) A storage including a non-volatile computer readable storage medium, the storage comprising:

a non-volatile memory storing a first inode locating a first file in said storage and also storing a journal comprising a list of committed inodes; and

a block manager configured to copy said first inode to a second inode, wherein said block manager is configured to change said second inode in response to updates to said first file, and wherein said block manager is configured to atomically update said first file in response to a commit of said first file by writing said second inode to said non-volatile memory, whereby said second inode locates said first file in said

storage, and wherein said block manager is configured to record said second inode in said journal.

3. (Original) The storage as recited in claim 2 wherein said commit of said first file comprises a commit command received from an external source which updates said first file.
4. (Original) The storage as recited in claim 3 wherein said commit command comprises a file close command.
5. (Original) The storage as recited in claim 3 wherein said commit command comprises an fsync command.
6. (Original) The storage as recited in claim 2 wherein said journal further includes a checkpoint record including a description of an inode file, a block allocation bitmap, and an inode allocation bitmap.
7. (Original) The storage as recited in claim 6 wherein the description comprises inodes for each of said inode file, said block allocation bitmap, and said inode allocation bitmap.
8. (Currently Amended) An apparatus comprising:
 - a computing node comprising at least one computer system, the computing node configured to perform one or more write commands to a file and a commit command committing the one or more write commands to said file; and
 - a storage including a non-volatile computer readable storage medium, the storage coupled to receive said one or more write commands and said commit command, wherein said storage is configured to copy one or more blocks of said file to a copied one or more blocks, said one or more blocks updated by said one or more write commands, and wherein said storage is configured to update said copied one or

more blocks with write data corresponding to said one or more write commands, and wherein said storage is configured to copy a first inode locating said file in said storage to a second inode and to update pointers within said second inode pointing to said one or more blocks to point to said copied one or more blocks, and wherein said storage is configured to atomically update said file by writing said second inode responsive to said commit command, and wherein said first inode is stored in an inode file, and wherein said inode file is identified by a master inode, and wherein said inode file is atomically updated with said second inode by writing said master inode subsequent to said commit command.

9. (Previously Presented) The apparatus as recited in claim 8 wherein said commit command comprises a file close command.

10. (Previously Presented) The apparatus as recited in claim 8 wherein said commit command comprises an fsync command.

11. (Cancelled)

12. (Previously Presented) A method comprising:

copying a first inode to a second inode, wherein said first inode locates a first file in a storage;

modifying said second inode in response to one or more changes to said first file; and

atomically updating said first file by establishing said second inode as the inode for said first file, wherein said establishing comprises storing said second inode in a journal stored in a nonvolatile memory.

13. (Original) The method as recited in claim 12 further comprising writing a master inode

corresponding to an inode file including said second inode to a checkpoint record in said journal.

14. (Original) The method as recited in claim 13 wherein recovering from a system failure comprises:

scanning said journal to locate a most recent checkpoint record and zero or more inodes subsequent to said most recent checkpoint record within said journal;

copying said master inode from said most recent checkpoint record to a volatile memory; and

updating an inode file corresponding to said master inode with said one or more inodes subsequent to said most recent checkpoint record.

15. (Original) The method as recited in claim 14 wherein said updating said inode file comprises:

copying one or more blocks of said inode file storing said one or more inodes to a copied one or more blocks; and

updating said master inode in said volatile memory to point to said copied one or more blocks.

16. (Previously Presented) The method as recited in claim 12 wherein said block map further comprises a first inode allocation bitmap indicating which inodes within said first inode file are allocated to files, the method further comprising:

copying said first inode allocation bitmap to a second inode allocation bitmap;

modifying said second inode allocation bitmap to reflect one or more inodes allocated to new files; and

establishing a third inode within said block map to said second inode allocation bitmap subsequent to said modifying said second inode bitmap.

17. (Previously Presented) The method as recited in claim 16 wherein said block map further comprises a first block allocation bitmap indicating which blocks within a storage including said block map are allocated to files, the method further comprising:

copying said first block allocation bitmap to a second block allocation bitmap;

modifying said second block allocation bitmap to reflect one or more blocks allocated to files; and

establishing a fourth inode within said block map to said second block allocation bitmap subsequent to said modifying said second block allocation bitmap.

18. (Previously Presented) The method as recited in claim 12 wherein said establishing said second inode is performed in response to a commit command.

19. (Original) The method as recited in claim 18 wherein said commit command is a close file command.

20. (Original) The method as recited in claim 18 wherein said commit command is an fsync command.

21. (Cancelled)

22. (Currently Amended) A storage including a non-volatile computer readable storage medium,

the storage comprising:

a non-volatile memory storing a first inode locating a first version of a file in said storage and also storing a journal comprising a list of committed inodes; and

a block manager configured to copy said first inode to a second inode, wherein said block manager is configured to change said second inode in response to updates to the file, and wherein said block manager is configured to atomically update the file, producing a second version of the file, in response to a commit of the file by writing said second inode to said non-volatile memory, wherein said second inode locates said second version of said file in said storage, and wherein said block manager is configured to record said second inode in said journal.

23. (Original) The storage as recited in claim 22 wherein said commit of the file comprises a commit command received from an external source which updates the file.

24. (Original) The storage as recited in claim 23 wherein said commit command comprises a file close command.

25. (Original) The storage as recited in claim 23 wherein said commit command comprises an fsync command.

26. (Original) The storage as recited in claim 22 wherein said journal further includes a checkpoint record including a description of an inode file, a block allocation bitmap, and an inode allocation bitmap.

27. (Original) The storage as recited in claim 26 wherein the description comprises inodes for each of said inode file, said block allocation bitmap, and said inode allocation bitmap.

28. (Cancelled)

29. (Previously Presented) A method comprising:

copying a first inode to a second inode, wherein said first inode locates a first version of a file in a storage;

modifying said second inode in response to one or more changes to the file, creating a second version of the file; and

atomically updating the file to the second version by establishing said second inode as the inode for the file, wherein said establishing comprises storing said second inode in a journal stored in a nonvolatile memory.

30. (Original) The method as recited in claim 29 further comprising writing a master inode corresponding to an inode file including said second inode to a checkpoint record in said journal.

31. (Original) The method as recited in claim 30 wherein recovering from a system failure comprises:

scanning said journal to locate a most recent checkpoint record and zero or more inodes subsequent to said most recent checkpoint record within said journal;

copying said master inode from said most recent checkpoint record to a volatile memory; and

updating an inode file corresponding to said master inode with said one or more inodes subsequent to said most recent checkpoint record.

32. (Original) The method as recited in claim 31 wherein said updating said inode file comprises:

copying one or more blocks of said inode file storing said one or more inodes to a copied one or more blocks; and

updating said master inode in said volatile memory to point to said copied one or more blocks.

33. (Previously Presented) The method as recited in claim 29 wherein said block map further comprises a first inode allocation bitmap indicating which inodes within said first inode file are allocated to files, the method further comprising:

copying said first inode allocation bitmap to a second inode allocation bitmap;

modifying said second inode allocation bitmap to reflect one or more inodes allocated to new files; and

establishing a third inode within said block map to said second inode allocation bitmap subsequent to said modifying said second inode bitmap.

34. (Previously Presented) The method as recited in claim 33 wherein said block map further comprises a first block allocation bitmap indicating which blocks within a storage including said block map are allocated to files, the method further comprising:

copying said first block allocation bitmap to a second block allocation bitmap;

modifying said second block allocation bitmap to reflect one or more blocks allocated to files; and

establishing a fourth inode within said block map to said second block allocation bitmap

subsequent to said modifying said second block allocation bitmap.

35. (Previously Presented) The method as recited in claim 29 wherein said establishing said second inode is performed in response to a commit command.

36-44. (Cancelled)

Response to Arguments

4. The Applicants' arguments presented in the Appeal Brief filed on July 14, 2008 have been fully considered and are persuasive.

Allowable Subject Matter

5. *Claims 2-10, 12-20, 22-27 and 29-35 are allowed. The claims indicated include limitations that the prior arts of record do not appear to teach or render obvious, hence they are allowed.*
6. The following is an examiner's statement of reasons for allowance:
- As presented in the previous Office Action, Kozakura (US005724581) discloses, "a current page table 2 is provided in the main storage unit and manages the position information in the data base storage unit concerning the latest physical page storing the latest updated data and the shadow physical page storing the data before the latest update" (Kozakura, col.4, lines 41-45) and "a current table management table 3 is provided in the main storage unit and manages as a shadow page table the current page table whose backup data are copied when a checkpoint is recorded, and manages the*

current page table updated after the checkpoint as the latest page table" (Kozakura, col.4, lines 46-51). Hence, Kozakura teaches of the current and shadow page tables (i.e., Applicant's inodes) storing the position information of the physical data base data. In addition, Kozakura discloses, *"the present invention comprises a current page table for storing a page table in which a shadow page system manages a physical page corresponding to each logical page in a data base, and a current page table management table for managing the page table in the current page table using the shadow page system"* (Kozakura, col.3, lines 34-39). Hence, Kozakura teaches of the current page table management table (i.e., Applicant's journal) for managing the page tables (i.e., Applicant's inodes). Furthermore, Kozakura discloses, *"a non-volatile semiconductor memory such as a flash memory, a RAM disk, etc. can be used as the secondary storage unit 40"* (Kozakura, col.20, lines 28-30). Hence, Kozakura teaches of using non-volatile memory to store the current and shadow page tables (i.e., Applicant's inodes), which, in turn, stores the position information of the physical data base data. However, the prior arts of record fail to teach or suggest individually or in combination as stated in the independent claims for *"a block manager configured to copy said first inode to a second inode, wherein said block manager is configured to change said second inode in response to updates to said first file, and wherein said block manager is configured to atomically update said first file in response to a commit of said first file by writing said second inode to said non-volatile memory, whereby said second inode locates said first file in said storage, and wherein said block manager is configured to record said second inode in said journal"* and in combination with other limitations as set forth in the independent claims, as well as Applicants' arguments presented on pages 6-18 of the Appeal Brief filed on January 17, 2008. In the fore mentioned amendment, the

Applicants argued, *"The Final Office Action mailed March 21, 2007 ("Office Action") alleges that the inodes are taught in Kozakura as the page tables. Appellant respectfully disagrees. Page tables locate physical pages stored in the memory system of the computer system, mapping logical pages used by the software to physical pages (see, e.g., Kozakura, col. 1, lines 32-28). Page tables that map logical pages to physical pages in memory have nothing to do with inodes that locate files on a storage. Page tables cannot anticipate inodes, as they are completely different and are used for different purposes. Additionally, logical and physical pages are fixed in size (see, e.g., Kozakura col. 1, line 34), whereas files can have any size. Appellant notes that the standard for anticipation is one of fairly strict identity. Kozakura's page tables are significantly different from inodes, and cannot anticipate them"* (pg.6-7). In addition, the allowance is based on the Board Of Appeals' decision dated September 23, 2009; in which the Board stated, *"Thus, we find the terms "inode" and "page table" are distinct, well- established terms of art, with an "inode" being directed to a low-level integral aspect of a file system, as described in Appellant's Specification (FF 1-2; i.e., the inode includes pointers to each block storing the file data and may also include various file attributes)"* (pg.8).

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Thomas Duong whose telephone number is 571/272-3911. The

Art Unit: 2445

examiner can normally be reached on M-F 7:30AM - 4:00PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Vivek Srivastava can be reached on 571/272-7304. The fax phone numbers for the organization where this application or proceeding is assigned are 571/273-8300 for regular communications and 571/273-8300 for After Final communications.

/Thomas Duong/

Patent Examiner, Art Unit 2445

February 5, 2010

/Rupal D. Dharia/

Supervisory Patent Examiner, Art Unit 2400